下载APP

18 | 增长模型：用XGBoost评估裂变海报的最佳受众群体

2021-10-08 黄佳

《零基础实战机器学习》 课程介绍 >

讲述：黄佳
时长 24:28 大小 22.42M

你好，我是黄佳。欢迎来到零基础实战机器学习。

上一讲中，我们通过多种集成学习方法，预测出了用户接收到某个裂变后是否会购买产品，这是二分类问题。这一讲中呢，我们继续研究裂变，不过，我们不再对用户是否转化进行预测，而是要评估裂变方案的最佳受众是谁。通过这一讲，我希望你能掌握一种解决运营问题的重要思路：增长模型。这个模型，常用于评估营销活动、推广海报对用户的订单增量的影响，国外的一些大厂也在使用它提升促销转化率。

这一讲内容有点小复杂，我们就开门见山，直接介绍运营部门面临的具体问题吧。

问题的定义和分析

运营团队为易速鲜花的用户设计了两张 H5 海报页面，对应着两种裂变方案，分别是“情侣花享”和“拼团盛放”。前者是类似于买一送一的促销，后者是用户生成专属海报拉朋友拼团，团越大折扣越大。

经过一段时间的试运营，运营团队收集到了下面这些数据给到我们，这个数据集你可以在这里下载。

	A	B	C	D	E	F	G	H	I	J
1	用户码	曾助力	曾拼团	曾推荐	设备	城市类型	R值	M值	裂变类型	是否转化
2	1	0	1	0	电脑	中小	10	138	情侣花享	0
3	2	0	1	0	电脑	一线	4	105.59	拼团盛放	0
4	3	0	1	0	电脑	一线	1	494.13	拼团盛放	1
5	4	0	1	1	手机	二线	10	148.45	没有促销	0
6	5	1	0	0	手机	二线	2	337.18	拼团盛放	0
7	6	0	1	1	电脑	二线	10	56.48	情侣花享	0
8	7	1	1	1	其它	一线	2	551.98	情侣花享	0

可以看到，这个数据集不仅包含了两种裂变方案的用户转化数据（也就是购买与否），还包含了在没有任何促销活动情况下，用户的转化情况。

这其实是运营人员常用的 A/B 测试，就是在其它条件完全相同的情况下，系统随机给三分之一的用户显示“情侣花享”页，给另外三分之一的用户显示“拼团盛放”页，然后给剩下的三分之一用户显示没有促销的页面，以此来观察两种促销活动（裂变方案）的转化率增量。所以，这里的“裂变类型”字段就是指该用户接收到的促销页面，是系统随机分配的结果。

现在，运营团队想知道，这两种裂变方案的转化率增量是怎样的？也就是说，这两种裂变方案对一个用户“购买可能性”（购买概率）的增量分别是多少？

这可能不太好理解，我给你举个例子。就拿“情侣花享”这个方案来说，如果一个 50 多岁的女性用户平时没有买花的需求，也不是“情侣花享”的潜在用户。那她在收到这个促销活动页面后，就不会产生购买的念头，也就是说“购买可能性”不会发生变化，增量为 0。那这时候我们也不用花精力对她做这样的促销了，因为这个方案在她身上不起作用。

同样的道理，如果一个用户正在犹豫：老婆即将过生日，要不要给买束鲜花呢？就在这时，啪！“情侣花享”的促销活动来了，那他很可能就当场下单了。这种情况，就是“情

侣花享”方案增加了该用户的“购买可能性”，具体的增量是多少，需要机器学习模型来预测。

当然，也会有这样一种情况：假设有一个 20 岁左右、已经是 3 年会员的某男性用户，是我们的忠粉，他每个月都会给女友买花。对于这样的用户，即便我们不做任何促销，他也会购买产品。所以，我们做不做裂变推广、采用哪一种裂变方案，都对他的最终购买决策没有任何影响。因此，这时候，我们就希望机器学习模型可以判断出，该用户购买可能性的增量为 0。那运营团队就不用花费力气把促销海报发给他，也不需要过多考虑他更喜欢哪种折扣方案。对吧？

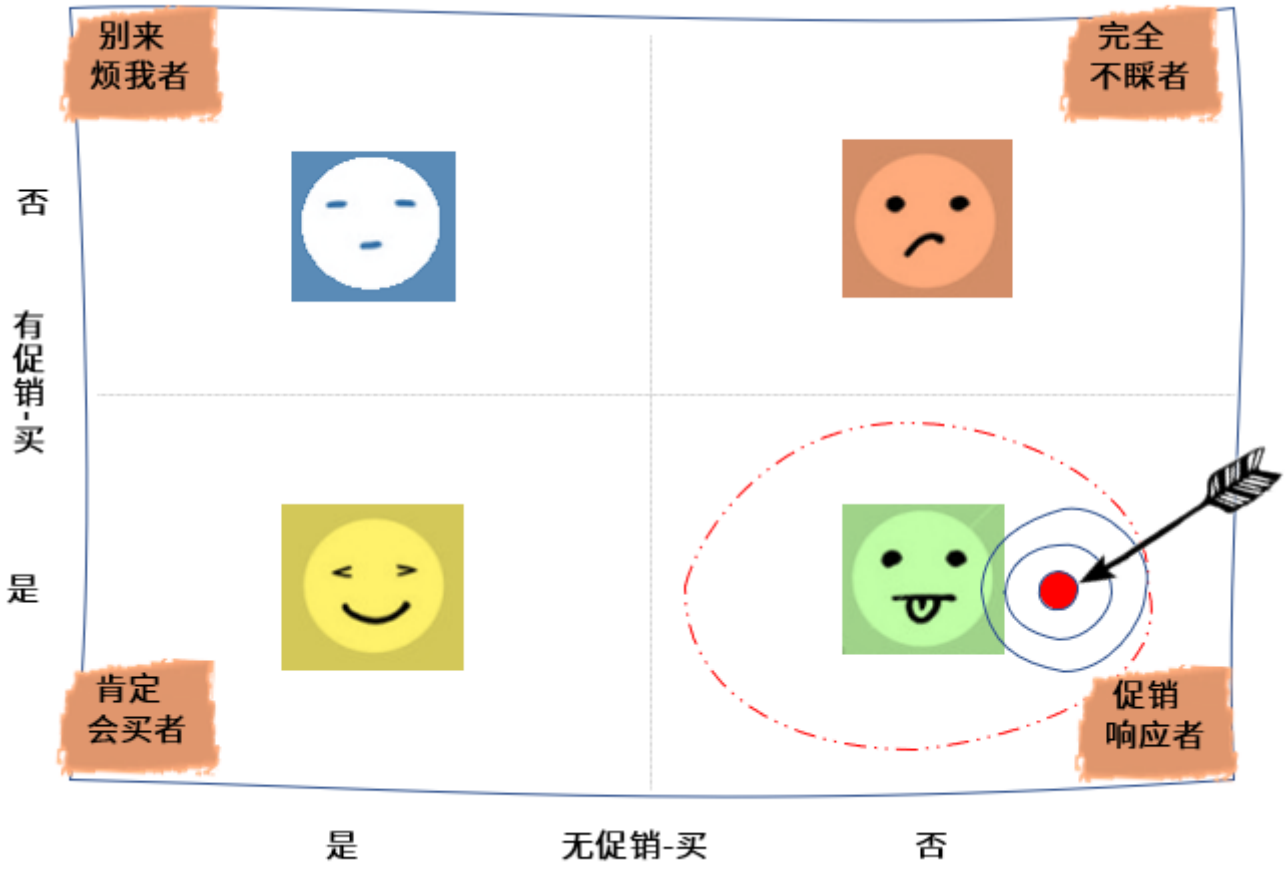
到这里，你应该清楚这个问题的含义了，我们实际上就是要预测：在一个特定裂变促销的前后，一个用户“购买可能性”的增量是多少？并且，你可能已经发现了，我们其实就是要从中找到那些正在犹豫“要不要买”的摇摆者，只有这些人，才**最值得**把特定的裂变方案发给他。

那我们怎么解决这个问题呢？这就需要增长模型大显身手了。

增长模型

增长模型，英文是 Uplift modelling 或者 incremental modelling，所以也翻译为提升建模、增量建模等，它多用于评估营销活动对用户的订单增量的影响。国外的互联网公司优步（Uber）等，也在使用该模型进行数据分析，以增加用户的下单次数。 请你注意哈，这里的增长模型是解决我们今天的问题的一种思路，它并不是一种具体的机器学习算法。

在介绍具体的思路之前，我们首先来了解一下，针对某个特定的裂变推广方案，用户可能有多少种不同的反应。前面几个例子就展示了三种反应，其实，总体来看，一共会有 4 种结果，也就是说，一个用户在接到促销之后，做出的反应必然属于下面这四种情况之一：



根据不同的反应，我们可以把用户分为四个类型：

第一类人是“完全不睬者”：这类人很容易理解，英文叫 Lost Causes，他们对当前促销产品没有兴趣，不管你怎么打广告，给什么样的优惠，裂变玩出花来，他们也不会买这个产品。可以说是一毛不拔，所以，在做促销时我们可以把他们忘掉。上面第一个例子的用户就属于这个群体。

第二类人是“促销响应者”：这类人是我们关注的重点，英文叫 Persuadable。他们只有在收到我们的促销活动后，才会购买。这类人对价格和折扣非常敏感，要等有合适的活动才肯花钱。就比如去京东买书，一看，只有 9 折？折扣力度不大，那就等“618”或者“双 11”吧，并且在离开网站之前，他们还可能在想买的商品中标注一个降价提醒。上面第二个例子中的用户，就是一个促销响应者。

第三类人是“肯定会买者”：这类人代表我们的忠实用户，英文叫 Sure Things。他们很满意当前产品，即便没有特别的促销活动也会购买。上面第三个例子中的用户，就属于这一类。

第四类人是“别来烦我者”：这一类人比较奇怪，英文叫 Do-Not-Disturb 或 Sleeping Dogs。这是一群收到促销反而不会购买的用户。他们内心的想法估计是：我最讨厌广告，你越想让我买，我就越不买！又或者是：你什么时候不来烦我，我再买。所以，你会惊讶地看到，这第四类人在有促销的时候不会购买，在无促销的时候反而会买。

而增长模型的原理就是要尽可能地定位出“促销响应者”和“别来烦我者”，来进行促销活动。从刚才的分析，我们可以知道，促销活动是 A 好还是 B 好，不是第一和第三类人说了算，而是第二和第四类人决定的。因为，促销活动会影响他们的决定。对于第二类人促销响应者，不打折他们就不会购买，所以促销活动就应该针对这个群体开展。与此同时，促销活动应该尽量避免发送到第四类人那里，因为促销会令他们反感。

不过，在实际项目中，我们是无法直接找出这四类人的。但我们可以通过“有没有收到裂变”和“是否转化”这两个字段，把整个用户做这样一个划分：裂变购买者、裂变未购买者、无裂变购买者和无裂变未购买者。请你注意，虽然我们在这里的划分，并不直接对应于上面讲的那“4 类人”，但是它们二者之间还是有关联的：

裂变购买者：包括“肯定会买者”和“促销响应者”；

裂变未购买者：包括“完全不睬者”和“别来烦我者”；

无裂变购买者：包括“肯定会买者”和“别来烦我者”；

无裂变未购买者：包括“完全不睬者”和“促销响应者”。

然后，我们可以通过机器学习模型来定位，每一个用户落入上面这 4 种人（也就是 4 种可能情况）的概率。再通过下面这个增量公式（这个公式的说明我们一会儿再细讲）求出“促销响应”的概率增加和“别来烦我”的概率减少，就是裂变所带来的转化增量了，即：

$$\text{增量分数} = （\text{裂变购买者} + \text{无裂变未购买者}） - （\text{裂变未购买者} + \text{无裂变购买者}）$$

因此，从整体上看，用增长模型解决我们这个项目问题的思路，可以分为两部分：首先是判断一个用户落入这 4 种分类的概率分别是多少；然后根据概率，通过增量公式来计算出一个用户在收到裂变页面前后的购买概率的增量。

那现在，我们就带着这个思路，进入这个项目的具体实战。

读入数据、预处理

还是一样，按照机器学习的步骤，我们先导入相关的包，并读入数据：

复制代码

```
import pandas as pd #导入Pandas
import numpy as np #导入NumPy
df_fission = pd.read_csv('易速鲜花增长模型.csv') #载入数据
print('用户数:', df_fission.count()['用户码']) #查看数据条目数
df_fission.head() #显示头几行数据
```

输出如下：

	用户码	曾助力	曾拼团	曾推荐	设备	城市类型	R值	M值	裂变类型	是否转化
0	1	0	1	0	电脑	中小	10	138.00	情侣花享	0
1	2	0	1	0	电脑	一线	4	105.59	拼团盛放	0
2	3	0	1	0	电脑	一线	1	494.13	拼团盛放	1
3	4	0	1	1	手机	二线	10	148.45	没有促销	0
4	5	1	0	0	手机	二线	2	337.18	拼团盛放	0

这个数据集中，共 64000 个数据样本，其中曾助力、曾拼团、曾推荐是该用户曾经参与过的活动的记录。而裂变类型字段就是随机分配给该用户的促销裂变页面。

下面，我们做一个简单的数据可视化，观察一下各用户群组对促销的转化率情况。

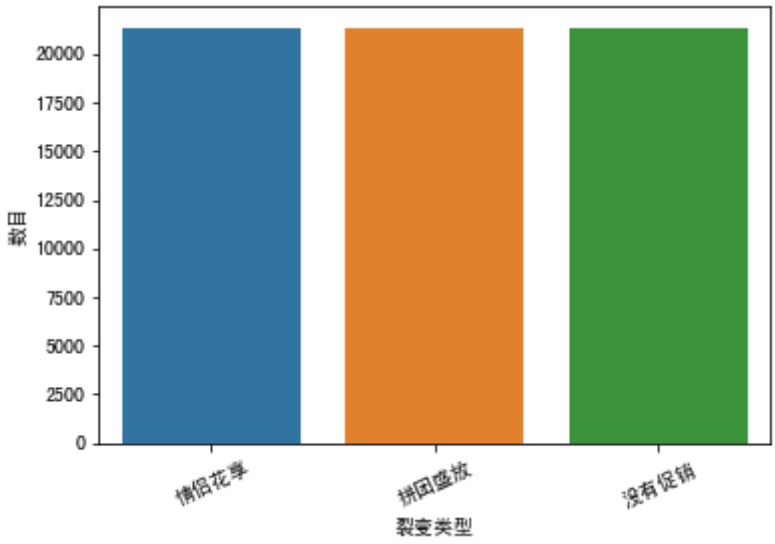
我们先来显示随机分配“情侣花享”、“拼团盛放"和“没有促销”的用户人数，看看收到这三种页面的用户数量是不是和我们最开始说的一样，大致相等。因为我们要用同等数量的数据样本来比较转化效果的差异。

复制代码

```
import matplotlib.pyplot as plt #导入pyplot模块
    import seaborn as sns #导入Seaborn
    fig = sns.countplot('裂变类型', data=df_fission) #创建柱状计数图
fig.set_xticklabels(fig.get_xticklabels(),rotation=25)    #X轴标签倾斜
fig.set_ylabel("数目") #Y轴标题
```

```
6 plt.show() #显示图像
```

输出如下：



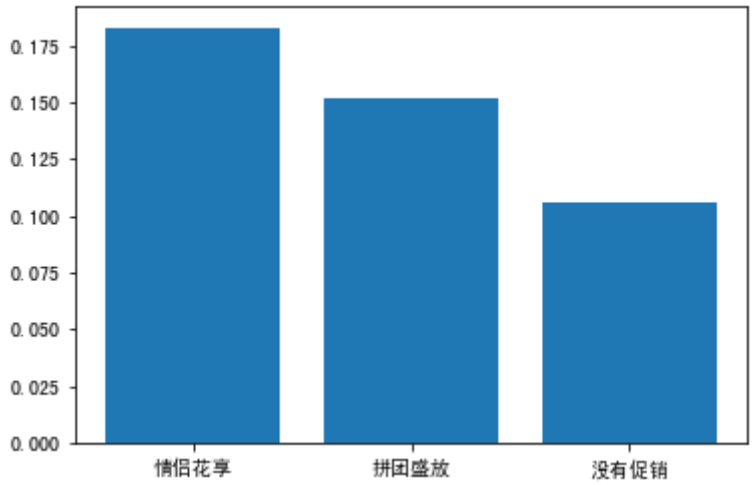
看来收到助力砍价、拼团狂买和无裂变页面的用户数是基本相同的，这和我们所期望的相符。

因为每一个用户是否成功转化都已记录在案，下面我们就来计算两种裂变方案带来的转化率，与无裂变情况下的转化相比，有没有提升，并显示出每一种情况下购买产品的用户所占的比例，也就是转化率的均值。

复制代码

```
1 df_plot = df_fission.groupby('裂变类型').是否转化.mean().reset_index() #促销分组的
2 plt.bar(df_plot['裂变类型'],df_plot['是否转化']) #不同促销转化均值柱状图
```

输出如下：



可以看出，“情侣花享”和“拼团盛放”这两种裂变模式，都能带来转化率的显著提升，但提升的效果是有差异的。

分析完数据后，我们接下来做特征工程，把原来的数据变为机器模型可读入的格式。这段代码是为机器学习模型创建哑变量，我们以前做过很多次，相信你已经很熟悉了。

复制代码

```
df_dummies = df_fission.drop(['裂变类型'],axis=1) #在拆分哑变量前，先拿掉裂变类型
df_dummies = pd.get_dummies(df_dummies) #为分类变量拆分哑变量
df_dummies['裂变类型'] = df_fission['裂变类型'] #把裂变类型放回去
df_fission = df_dummies.copy() #把哑变量数据集复制给元数据集
df_fission.head() #显示数据
```

输出如下：

	用户码	曾助力	曾拼团	曾推荐	R值	M值	是否转化	设备_手机	设备_电脑	城市类型_中小	城市类型_二线	裂变类型
0	1	0	1	0	10	138.00	0	0	1	1	0	情侣花享
1	2	0	1	0	4	105.59	0	0	1	0	0	拼团盛放
2	3	0	1	0	1	494.13	1	0	1	0	0	拼团盛放
3	4	0	1	1	10	148.45	0	1	0	0	1	没有促销
4	5	1	0	0	2	337.18	0	1	0	0	1	拼团盛放

经过前面的分析我们知道，要想知道在一个特定的裂变方案，对一个用户“购买可能性”的影响有多大，就需要把该裂变方案与无裂变的情况做独立比较，才能得到该裂变方案带来的购买可能性的增量（转化增量）。所以，我们先聚焦于“情侣花享”这种裂变模式，通过下面的代码，找出"情侣花享"和“没有促销”这种两组用户的所有数据：

复制代码

```
df_discount = df_fission.query("裂变类型 == '情侣花享' | 裂变类型 == '没有促销'") #
```

得到了这两组用户数据后，我们就可以开始构建这个数据集的标签了。

构建增长标签

我们前面分析到，要先把用户划分为四类：裂变购买者、裂变未购买者、无裂变购买者和无裂变未购买者。根据用户的一系列特征，判断一个用户落入这 4 种用户群组的概率分别是多少。所以，这 4 个用户群组就是我们的增长标签。我们用 0，1，2，3 来给每一类群组的标签编码：

0 代表裂变购买者；

1 代表裂变未购买者；

2 代表无裂变购买者；

3 代表无裂变未购买者。

然后，我们通过下面的代码，构建增长模型的标签：

复制代码

```
df_discount.loc[(df_discount.裂变类型 == '情侣花享') & (df_discount.是否转化 == 1)
df_discount.loc[(df_discount.裂变类型 == '情侣花享') & (df_discount.是否转化 == 0)
df_discount.loc[(df_discount.裂变类型 == '没有促销') &  (df_discount.是否转化 == 1
df_discount.loc[(df_discount.裂变类型 == '没有促销') &  (df_discount.是否转化 == 0
df_discount.head()
```

输出如下：

	用户码	曾助力	曾拼团	曾推荐	R值	M值	是否转化	设备_手机	设备_电脑	城市类型_中小	城市类型_二线	裂变类型	标签
0	1	0	1	0	10	138.00	0	0	1	1	0	情侣花享	1.0
3	4	0	1	1	10	148.45	0	1	0	0	1	没有促销	3.0
5	6	0	1	1	10	56.48	0	0	1	0	1	情侣花享	1.0
6	7	1	1	1	2	551.98	0	0	0	0	0	情侣花享	1.0
8	9	1	0	1	7	29.99	1	0	1	0	1	情侣花享	0.0

有了分类标签之后，我们接下来就构建出特征集和标签集，并进行训练集和测试集地拆分：

```
复制代码
1 X = df_discount.drop(['标签','是否转化'],axis=1) #特征集，Drop掉便签相关字段
2 y = df_discount.标签 #标签集
3 from sklearn.model_selection import train_test_split
4 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rando
```

接着，我们进入算法选择环节。

选择算法训练模型

判断一个用户落入 4 种分类的概率分别是多少，这其实是一个典型的多类别分类问题。解决多类别分类问题的算法和二分类问题的算法完全相同，可以通过一系列传统机器学习算法或者深度学习神经网络算法来完成。这里我们就选择 XGBoost 算法。

我之所以会选择 XGBoost 算法求分类概率，是因为在上一讲中，我们已经看到 XGBoost 算法作为一种集成学习方法，在分类问题中具有超强的性能。 XGBoost 算法，可以说是目前最有名且最实用的集成学习算法了。

下面，我们就导入、创建并拟合 xgboost 模型：

```
复制代码
1 import xgboost as xgb #导入xgboost模型
2 xgb_model = xgb.XGBClassifier() #创建xgboost模型
3 xgb_model.fit(X_train.drop(['用户码','裂变类型'], axis=1), y_train) #拟合xgboost模
```

模型训练成功之后，我们就可以计算出每一个用户落入 4 种分类标签的概率了。

```
复制代码
1 uplift_probs = xgb_model.predict_proba(X_test.drop(['用户码','裂变类型'], axis=1
2 uplift_probs #显示4种概率
```

输出如下：

复制代码

```
array([[0.00787458, 0.51820630, 0.04026253, 0.43365657],
       [0.04618792, 0.46886584, 0.01700269, 0.46794358],
       [0.06478575, 0.5077343 , 0.06103172, 0.36644822],
       ...,
       [0.11078195, 0.40745038, 0.06231715, 0.4194505 ],
       [0.17259875, 0.38025224, 0.11393689, 0.33321217],
       [0.06996422, 0.33474946, 0.03079985, 0.5644865 ]], dtype=float32)
```

注意，这里我们并不需要用 predict 函数直接求得分类标签的值（0，1，2，3），因为这些值计算不出裂变增量，而是通过 predict_proba，得到了测试集中每一个用户落入 4 种分类标签的概率数组。其中，哪个标签的概率大，就说明用户落入其中可能性大。

比如说，第三行数据[0.00787458，0.5182063，0.04026253，0.43365657]，这个 4 个概率分别代表裂变购买者、裂变未购买者、无裂变购买者和无裂变未购买者。也就是说，XGBoost 模型预测这个用户有 0.5182063 的可能性成为裂变购买者，有 0.00787458 的可能性成为裂变未购买者，有 0.04026253 成为无裂变购买者，0.43365657 成为无裂变未购买者。

下面，我们就进入了关键环节：计算特定裂变所带来的具体增量。

计算裂变增量

上面的机器学习模型已根据数据对每一个用户的购物特点作出预测。那么，对于每一种裂变方案，我们应该如何算出增量的大小呢？我们前面说可以用增量公式来计算：

$$增量分数 = （裂变购买者 + 无裂变未购买者） - （裂变未购买者 + 无裂变购买者）$$

那么上面这个增量分数的计算公式是怎么推出来的呢？其实，原始的增量公式（Generalized Weighed Uplift）是这样的：

$$增量分数 = \frac{P(TR)}{P(T)} + \frac{P(CN)}{P(C)} - \frac{P(TN)}{P(T)} - \frac{P(CR)}{P(C)}$$

注意，增量分数就是指增量的大小，在这个公式中：

TR，即 treatment responsive，表示“有促销应答”，对应于“裂变购买者”；

TN，即 treatment non-responsive，表示“无促销应答”，对应于“裂变未购买者”；

CR，即 control responsive，表示“有控制应答”，对应于“无裂变购买者”；

CN，即 control non-responsive，表示“无控制应答”，对应于“无裂变未购买者”。

而公式里的 P 表示概率值，T 表示促销人群 (TR + TN)，C 表示无促销人群 (CR + CN)。关于这个原始的增量公式，如果你有兴趣想深入了解，可以看看这篇论文。

在这次实战的数据分配中，因为控制组和裂变组的人数是相等的，公式中的分母 P(T) 和 P(C) 都是一个定值，即 50%。所以，这个公式也可以简化为：

$$增量分数 = (P(TR) + P(CN)) - (P(TN) + P(CR))$$

由于我们的目标是要定位“**促销响应者**”和“**别来烦我者**”这两种人，**其实也就是想求出有裂变较之无裂变所带来的促销响应可能性的提升和别来烦我可能性的减少**，所以，你可以这样理解上面的增量分数公式：

$$增量分数 = （裂变购买者 + 无裂变未购买者） - （裂变未购买者 + 无裂变购买者）$$

或者也可以写成：

$$增量分数 = (促销响应可能性的提升) + (别来烦我可能性的减少)$$

公式的前半部分 (P(TR) + P(CN)) 就包含促销响应者，虽然也有肯定会买者和完全不睬者，但无需关注。后半部分 P(TN) + P(CR) 则代表着别来烦我者，所以要取负值，这类人越少越好，其中也有肯定会买者和完全不睬者，但也无需关注。

因此，根据上面的增量公式，就求出了每个用户裂变促销前后的“购买可能性”的增量。

下面的代码就求出了这种裂变为每一个用户带来的增量分数。

复制代码

```
discount_uplift = X_test.copy() #构建增量分数数据集
discount_uplift['P_TR'] = uplift_probs[:,0] #添加有应答裂变概率
discount_uplift['P_TN'] = uplift_probs[:,1] #添加无应答裂变概率
discount_uplift['P_CR'] = uplift_probs[:,2] #添加有应答控制概率
discount_uplift['P_CN'] = uplift_probs[:,3] #添加无应答控制概率
#计算增量分数
discount_uplift['增量分数'] = discount_uplift.eval('P_TR + P_CN - (P_TN + P_CR)')
discount_uplift #显示增量分数
```

输出如下：

用户码	曾助力	曾拼团	曾推荐	R值	M值	设备_手机	设备_电脑	城市类型_中小	城市类型_二线	裂变类型	P_TR	P_TN	P_CR	P_CN	增量分数
50707	0	1	0	10	271.77	1	0	0	0	没有促销	0.104083	0.428877	0.020903	0.446138	0.100441
37014	0	1	0	3	250.76	0	1	0	0	没有促销	0.135687	0.473821	0.091070	0.299422	-0.129782
14686	0	1	0	2	341.75	1	0	0	1	没有促销	0.082415	0.399597	0.109551	0.408438	-0.018296
19738	0	1	1	10	29.99	0	1	0	1	没有促销	0.062630	0.467461	0.026409	0.443500	0.012260

这个分数，也就是用户从没有促销，到收到这个裂变方案的概率提升值。那么，这时我们如果按照相同的方法求出另外一中裂变方案的概率提升值，就能够比较出，对于一个特定的用户，哪种裂变形式，会更奏效！

而且，有了两种裂变方案针对于每一个用户的增量，我们还可以求出其均值。这个值就能用来比较两种裂变促销的效果优劣。也就是，整体上对于该 App 的用户来说，哪种裂变能够带来更多的增量。这样，我们今天的问题就得到了很好的解决。

总结一下

在这次实战中，我们解决了运营团队提出的，评估裂变方案适合哪类受众群体这个问题。这个解决方案的思路是通过比较两种裂变带来的转化率增长，同时看看各用户分组对裂变转化率造成的影响。

这个过程的核心，是根据机器学习模型来辅助判断每个用户是否适合推送当前裂变方案。首先调整数据集，只保留一种裂变类型（情侣花享）的用户组和无裂变用户组，并根据用户组别以及是否购买商品这两个字段，为数据集中的每一个用户构建出标签，分别是裂变购买者 TR、裂变未购买者 TN、无裂变购买者 CR 和无裂变未购买者 CN。

然后，我们构建出机器学习的训练集和预测集，听过 XGBoost 算法创建机器学习模型，并根据训练集的特征和标签拟合模型。 拟合好的模型就可以在测试集上对其他用户进行预测，得出新用户落于上述四种情况的可能性，即 P(TR)、P(TN)、P(CR)、P(CN) 四个概率，并根据概率，计算裂变带来的增量。

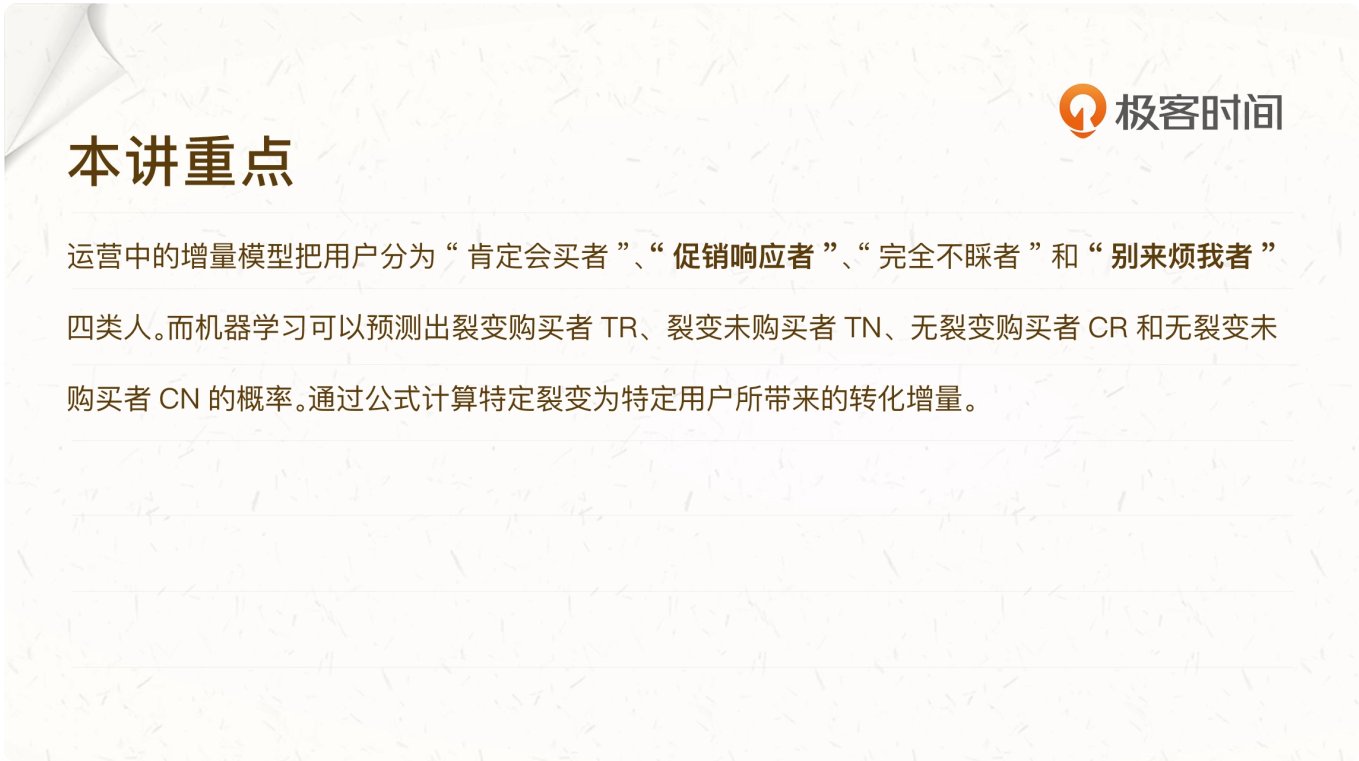
这个过程，我们灵活的应用了机器学习模型预测多分类问题的概率的功能，完成了对裂变方案的评估。希望这个 Case，能够启发你将机器学习模型应用到更多的业务运营场景中去。

思考题

我在这里给你留一个思考题。

我们在实战过程中，选择了“情侣花享”这种裂变形式，计算出引入该裂变之后的增量。请你继续选择“拼团盛放”这个裂变形式，完成类似的增量预测，然后比较一下两种裂变的增量均值，那个整体效果更好。

欢迎你在留言区和我分享你的观点，如果你认为这节课的内容有收获，也欢迎把它分享给你的朋友，我们下一讲再见！


极客时间
本讲重点
运营中的增量模型把用户分为“ 肯定会买者 ”、“ 促销响应者 ”、“ 完全不睬者 ”和“ 别来烦我者 ”四类人。而机器学习可以预测出裂变购买者 TR、裂变未购买者 TN、无裂变购买者 CR 和无裂变未购买者 CN 的概率。通过公式计算特定裂变为特定用户所带来的转化增量。

分享给需要的人，Ta订阅后你可得 20 元现金奖励

生成海报并分享

赞 1 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 17 | 集成学习：机器学习模型如何“博采众长”?

下一篇 19 | 胸有成竹：如何快速定位合适的机器学习算法？

精选留言

写留言

由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。